

# **Отчёт по лабораторной работе №5**

**Анализ файловой системы Linux. Команды для работы с файлами и каталогами**

Щербак Маргарита Романовна

2022

# 1 Цель работы:

Ознакомление с файловой системой Linux, её структурой, именами и содержанием каталогов. Приобретение практических навыков по применению команд для работы с файлами и каталогами, по управлению процессами (и работами), по проверке использования диска и обслуживанию файловой системы.

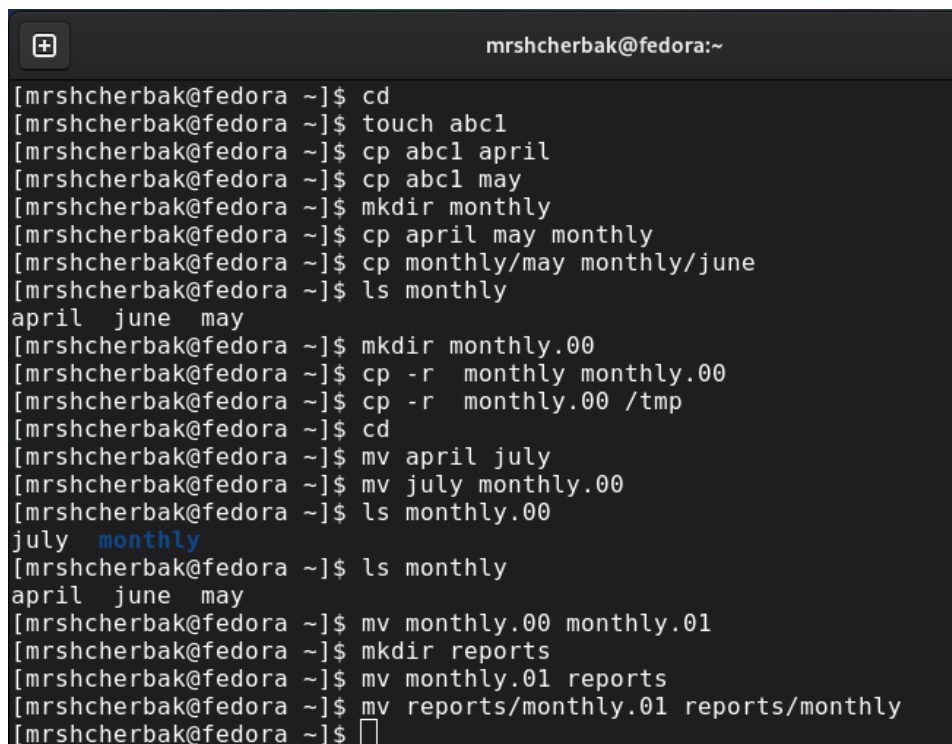
## 2 Теоретическое введение:

Для создания текстового файла можно использовать команду `touch`. Для просмотра файлов небольшого размера можно использовать команду `cat`. Для просмотра файлов постранично удобнее использовать команду `less`. Команда `cp` используется для копирования файлов и каталогов. Опция `i` в команде `cp` выведет на экран запрос подтверждения о перезаписи файла. Для рекурсивного копирования каталогов, содержащих файлы, используется команда `cp` с опцией `-r`. Команды `mv` и `mkdir` предназначены для перемещения и переименования файлов и каталогов. Каждый файл или каталог имеет права доступа. В сведениях о файле или каталоге указываются:

- тип файла (символ `-`) обозначает файл, а символ `d` — каталог);
- права для владельца файла (`r` — разрешено чтение, `w` — разрешена запись, `x` — разрешено выполнение, `-` — право доступа отсутствует);
- права для членов группы (`r` — разрешено чтение, `w` — разрешена запись, `x` — разрешено выполнение, `-` — право доступа отсутствует);
- права для всех остальных (`r` — разрешено чтение, `w` — разрешена запись, `x` — разрешено выполнение, `-` — право доступа отсутствует). Права доступа к файлу или каталогу можно изменить, воспользовавшись командой `chmod`. Сделать это может владелец файла (или каталога) или пользователь с правами администратора.

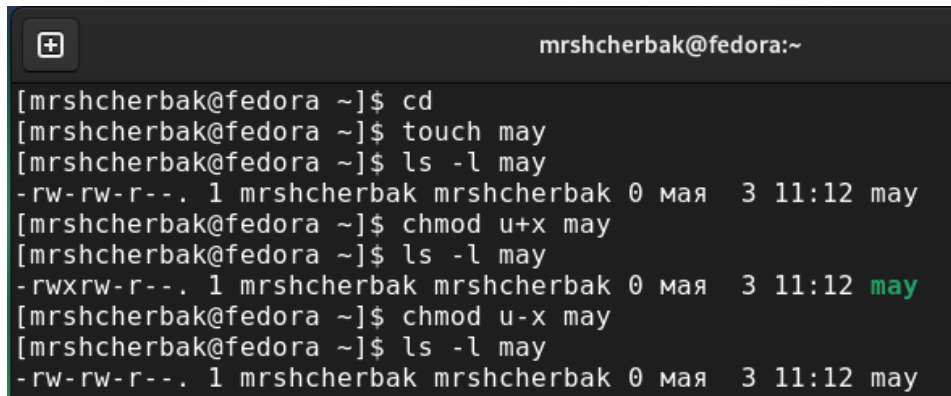
### 3 Выполнение лабораторной работы:

1. Выполнили все примеры, приведённые в первой части описания лабораторной работы. (Рис. 3.1 - Рис. 3.3)



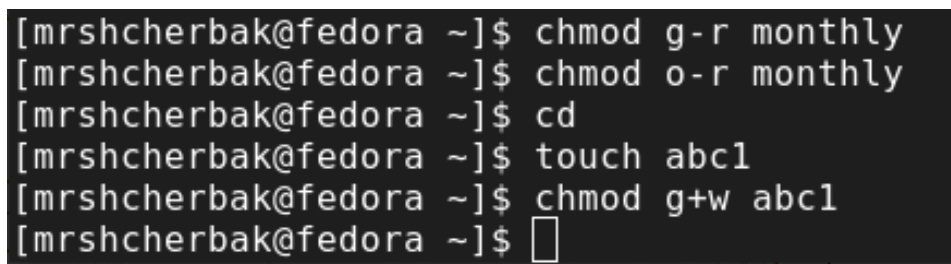
```
mrshcherbak@fedora:~  
[mrshcherbak@fedora ~]$ cd  
[mrshcherbak@fedora ~]$ touch abc1  
[mrshcherbak@fedora ~]$ cp abc1 april  
[mrshcherbak@fedora ~]$ cp abc1 may  
[mrshcherbak@fedora ~]$ mkdir monthly  
[mrshcherbak@fedora ~]$ cp april may monthly  
[mrshcherbak@fedora ~]$ cp monthly/may monthly/june  
[mrshcherbak@fedora ~]$ ls monthly  
april  june  may  
[mrshcherbak@fedora ~]$ mkdir monthly.00  
[mrshcherbak@fedora ~]$ cp -r monthly monthly.00  
[mrshcherbak@fedora ~]$ cp -r monthly.00 /tmp  
[mrshcherbak@fedora ~]$ cd  
[mrshcherbak@fedora ~]$ mv april july  
[mrshcherbak@fedora ~]$ mv july monthly.00  
[mrshcherbak@fedora ~]$ ls monthly.00  
july  monthly  
[mrshcherbak@fedora ~]$ ls monthly  
april  june  may  
[mrshcherbak@fedora ~]$ mv monthly.00 monthly.01  
[mrshcherbak@fedora ~]$ mkdir reports  
[mrshcherbak@fedora ~]$ mv monthly.01 reports  
[mrshcherbak@fedora ~]$ mv reports/monthly.01 reports/monthly  
[mrshcherbak@fedora ~]$
```

Рис. 3.1: Выполнение примеров



```
mrshcherbak@fedora:~  
[mrshcherbak@fedora ~]$ cd  
[mrshcherbak@fedora ~]$ touch may  
[mrshcherbak@fedora ~]$ ls -l may  
-rw-rw-r--. 1 mrshcherbak mrshcherbak 0 мая 3 11:12 may  
[mrshcherbak@fedora ~]$ chmod u+x may  
[mrshcherbak@fedora ~]$ ls -l may  
-rwxrw-r--. 1 mrshcherbak mrshcherbak 0 мая 3 11:12 may  
[mrshcherbak@fedora ~]$ chmod u-x may  
[mrshcherbak@fedora ~]$ ls -l may  
-rw-rw-r--. 1 mrshcherbak mrshcherbak 0 мая 3 11:12 may
```

Рис. 3.2: Примеры



```
[mrshcherbak@fedora ~]$ chmod g-r monthly  
[mrshcherbak@fedora ~]$ chmod o-r monthly  
[mrshcherbak@fedora ~]$ cd  
[mrshcherbak@fedora ~]$ touch abc1  
[mrshcherbak@fedora ~]$ chmod g+w abc1  
[mrshcherbak@fedora ~]$
```

Рис. 3.3: Команды

2. Выполнили следующие действия (Рис. 3.4).

- Скопировали файл `/usr/include/sys/io.h` в домашний каталог и назвали его `equipment`.
- В домашнем каталоге создали директорию `~/ski.places`.
- Переместили файл `equipment` в каталог `~/ski.places`.
- Переименовали файл `~/ski.places/equipment` в `~/ski.places/equiplist`.
- Создали в домашнем каталоге файл `abc1` и скопировали его в каталог `~/ski.places`, назвали его `equiplist2`.
- Создали каталог с именем `equipment` в каталоге `~/ski.places`.
- Переместили файлы `~/ski.places/equiplist` и `equiplist2` в каталог `~/ski.places/equipment`.
- Создали и переместили каталог `~/newdir` в каталог `~/ski.places` и назвали его `plans`.

```
mrshcherbak@fedora:~  
[mrshcherbak@fedora ~]$ cp /usr/include/sys/io.h /home/mrshcherbak/equipment  
[mrshcherbak@fedora ~]$ mkdir ski.plases  
[mrshcherbak@fedora ~]$ mv equipment ski.plases  
[mrshcherbak@fedora ~]$ mv /ski.plases/equipment /ski.plases/equiplist  
mv: не удалось выполнить stat для '/ski.plases/equipment': Нет такого файла или каталога  
[mrshcherbak@fedora ~]$ mv ski.plases/equipment ski.plases/equiplist  
[mrshcherbak@fedora ~]$ touch abcl  
[mrshcherbak@fedora ~]$ cp abcl ski.plases  
[mrshcherbak@fedora ~]$ mv ski.plases/abcl ski.plases/equiplist2  
[mrshcherbak@fedora ~]$ mkdir ski.plases/equipment  
[mrshcherbak@fedora ~]$ mv ski.plases/equiplist ski.plases/equipment  
[mrshcherbak@fedora ~]$ mv ski.plases/equiplist2 ski.plases/equipment  
[mrshcherbak@fedora ~]$ mkdir newdir  
[mrshcherbak@fedora ~]$ mv newdir ski.plases  
[mrshcherbak@fedora ~]$ mv ski.plases/newdir ski.plases/plans  
mv: не удалось выполнить stat для 'ski.plases/newdir': Нет такого файла или каталога  
[mrshcherbak@fedora ~]$ mv ski.plases/newdir ski.plases/plans  
[mrshcherbak@fedora ~]$
```

Рис. 3.4: Выполнение действий

3. Определили опции команды `chmod`, необходимые для того, чтобы присвоить перечисленным ниже файлам выделенные права доступа, считая, что в начале таких прав нет: (Рис. 3.5 - Рис. 3.8).

1. `drwxr-r- ... australia`
2. `drwx-x-x ... play`
3. `-r-xr-r- ... my_os`
4. `-rw-rw-r- ... feathers`

При необходимости создали нужные файлы.

```
[mrshcherbak@fedora ~]$ chmod 744 australia  
[mrshcherbak@fedora ~]$ ls -l  
итого 0  
-rw-rw-r--. 1 mrshcherbak mrshcherbak 0 мая 3 11:40 abcl  
drwxr--r--. 1 mrshcherbak mrshcherbak 0 мая 3 11:57 australia
```

Рис. 3.5: Владельцу каталога `australia` устанавливаем права доступа на чтение, запись и выполнение, группе людей и остальным пользователям - только на чтение

```

[mrshcherbak@fedora ~]$ cd
[mrshcherbak@fedora ~]$ mkdir play
[mrshcherbak@fedora ~]$ chmod 711 play
[mrshcherbak@fedora ~]$ ls -l
итого 0
-rw-rw-r--. 1 mrshcherbak mrshcherbak 0 мая 3 11:40 abc1
drwxr--r--. 1 mrshcherbak mrshcherbak 0 мая 3 11:57 australia
drwxr-xr-x. 1 mrshcherbak mrshcherbak 8 апр 29 23:40 bin
-rw-rw-r--. 1 mrshcherbak mrshcherbak 0 мая 3 11:12 may
drwx-wx--x. 1 mrshcherbak mrshcherbak 24 мая 3 10:54 monthly
drwx--x--x. 1 mrshcherbak mrshcherbak 0 мая 3 12:05 play

```

Рис. 3.6: Владельцу каталога play устанавливаем права доступа на чтение и запись, группе и остальным пользователям - только на выполнение

```

[mrshcherbak@fedora ~]$ touch my_os
[mrshcherbak@fedora ~]$ chmod 544 my_os
[mrshcherbak@fedora ~]$ ls -l
итого 0
-rw-rw-r--. 1 mrshcherbak mrshcherbak 0 мая 3 11:40 abc1
drwxr--r--. 1 mrshcherbak mrshcherbak 0 мая 3 11:57 australia
drwxr-xr-x. 1 mrshcherbak mrshcherbak 8 апр 29 23:40 bin
-rw-rw-r--. 1 mrshcherbak mrshcherbak 0 мая 3 11:12 may
drwx-wx--x. 1 mrshcherbak mrshcherbak 24 мая 3 10:54 monthly
-r-xr--r--. 1 mrshcherbak mrshcherbak 0 мая 3 12:06 my_os

```

Рис. 3.7: Владельцу файла my\_os устанавливаем права доступа на чтение и выполнение, группе и остальным пользователям - только на чтение

```

[mrshcherbak@fedora ~]$ touch feathers
[mrshcherbak@fedora ~]$ chmod 664 feathers
[mrshcherbak@fedora ~]$ ls -l
итого 0
-rw-rw-r--. 1 mrshcherbak mrshcherbak 0 мая 3 11:40 abc1
drwxr--r--. 1 mrshcherbak mrshcherbak 0 мая 3 11:57 australia
drwxr-xr-x. 1 mrshcherbak mrshcherbak 8 апр 29 23:40 bin
-rw-rw-r--. 1 mrshcherbak mrshcherbak 0 мая 3 12:08 feathers

```

Рис. 3.8: Владельцу и группе, к которой принадлежит владелец, файла feathers устанавливаем права доступа на чтение и запись, остальным пользователям - только на чтение

4. Проделали приведённые ниже упражнения (Рис. 3.9 - Рис. 3.10):

- Просмотрели содержимое файла /etc/passwd.

- Скопировали файл /feathers в файл /file.old.
- Переместили файл /file.old в каталог /play.
- Скопировали каталог /play в каталог /fun.
- Переместили каталог /fun в каталог /play и назвали его games.
- Лишили владельца файла ~/feathers права на чтение. Если я попытаюсь просмотреть файл ~/feathers командой cat, то мне будет отказано в доступе, так как у меня нет прав на чтение. Если я попытаюсь скопировать файл ~/feathers, то мне также будет отказано в доступе, так как у меня нет права на копирование. (r)
- Дали владельцу файла ~/feathers право на чтение.
- Лишили владельца каталога ~/play права на выполнение.
- Першли в каталог ~/play. В действии было отказано, так как у меня нет права на выполнение (x).
- Дали владельцу каталога ~/play право на выполнение.

```
[mrshcherbak@fedora etc]$ cd
[mrshcherbak@fedora ~]$ cp feathers file.old
[mrshcherbak@fedora ~]$ mkdir play
mkdir: невозможно создать каталог «play»: Файл существует
[mrshcherbak@fedora ~]$ mv file.old play
[mrshcherbak@fedora ~]$ cp -r play fun
[mrshcherbak@fedora ~]$ mv fun play
[mrshcherbak@fedora ~]$ cd play
[mrshcherbak@fedora play]$ mv fun games
```

Рис. 3.9: Выполнение действий



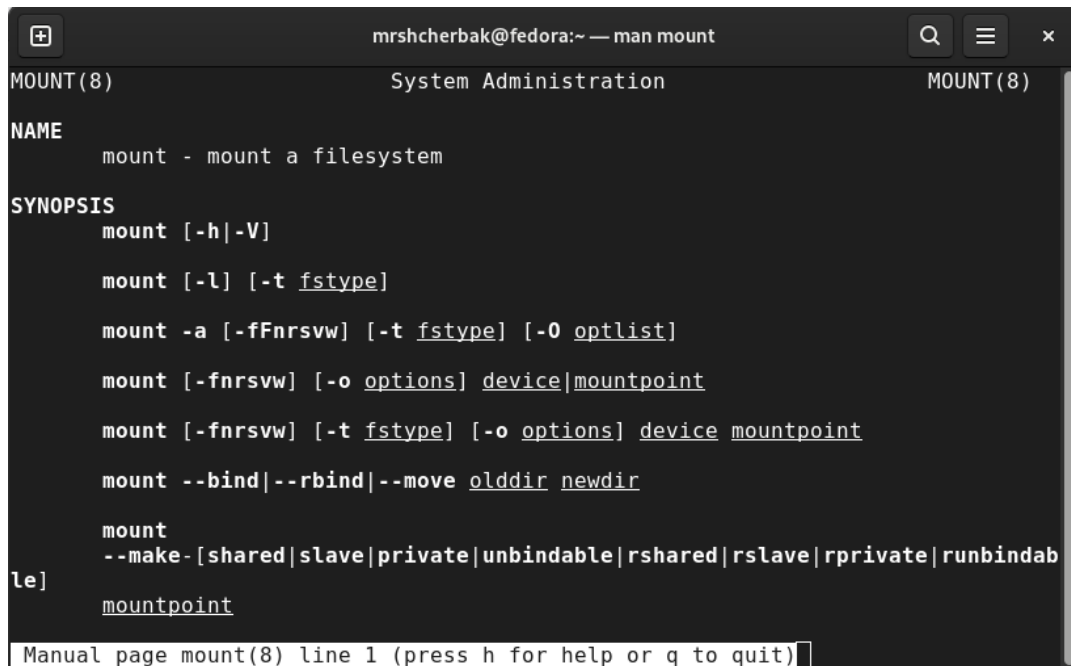
```

[mrshcherbak@fedora ~]$ ls
abcl      may      reports  Видео      Музыка
australia monthly  ski.plases  Документы  Общедоступные
bin       my_os    vvvvvvvvvv  Загрузки   'Рабочий стол'
feathers   play     work       Изображения  Шаблоны
[mrshcherbak@fedora ~]$ chmod u-r feathers
[mrshcherbak@fedora ~]$ cat feathers
cat: feathers: Отказано в доступе
[mrshcherbak@fedora ~]$ cp feathers feathers1
cp: невозможно открыть 'feathers' для чтения: Отказано в доступе
[mrshcherbak@fedora ~]$ chmod u+r feathers
[mrshcherbak@fedora ~]$ chmod u-x play
[mrshcherbak@fedora ~]$ cd play
bash: cd: play: Отказано в доступе
[mrshcherbak@fedora ~]$ chmod u+x play
[mrshcherbak@fedora ~]$ 

```

Рис. 3.10: Выполнение действий

5. Прочитали man по командам mount,fsck,mkfs,kill (Рис. 3.11 - Рис. 3.14).



```

MOUNT(8)                                System Administration                                MOUNT(8)

NAME
    mount - mount a filesystem

SYNOPSIS
    mount [-h|-V]

    mount [-l] [-t fstype]

    mount -a [-fFnrsvw] [-t fstype] [-O optlist]

    mount [-fnrsvw] [-o options] device|mountpoint

    mount [-fnrsvw] [-t fstype] [-o options] device mountpoint

    mount --bind|--rbind|--move olddir newdir

    mount
    --make-[shared|slave|private|unbindable|rshared|rslave|rprivate|runbindable]
mountpoint

Manual page mount(8) line 1 (press h for help or q to quit)

```

Рис. 3.11: man mount

```
mrshcherbak@fedora:~ — man fsck
FCK(8) System Administration FCK(8)

NAME
    fsck - check and repair a Linux filesystem

SYNOPSIS
    fsck [-lsAVRTMNP] [-r [fd]] [-C [fd]] [-t fstype] [filesystem...] [--]
    [fs-specific-options]

DESCRIPTION
    fsck is used to check and optionally repair one or more Linux
    filesystems. filesystem can be a device name (e.g., /dev/hdc1,
    /dev/sdb2), a mount point (e.g., /, /usr, /home), or an filesystem
    label or UUID specifier (e.g.,
    UUID=8868abf6-88c5-4a83-98b8-bfc24057f7bd or LABEL=root). Normally, the
    fsck program will try to handle filesystems on different physical disk
    drives in parallel to reduce the total amount of time needed to check
    all of them.

    If no filesystems are specified on the command line, and the -A option
    is not specified, fsck will default to checking filesystems in
    /etc/fstab serially. This is equivalent to the -As options.

Manual page fsck(8) line 1 (press h for help or q to quit)
```

Рис. 3.12: man fsck

```
mrshcherbak@fedora:~ — man mkfs
MKFS(8) System Administration MKFS(8)

NAME
    mkfs - build a Linux filesystem

SYNOPSIS
    mkfs [options] [-t type] [fs-options] device [size]

DESCRIPTION
    This mkfs frontend is deprecated in favour of filesystem specific
    mkfs.<type> utils.

    mkfs is used to build a Linux filesystem on a device, usually a hard
    disk partition. The device argument is either the device name (e.g.,
    /dev/hda1, /dev/sdb2), or a regular file that shall contain the
    filesystem. The size argument is the number of blocks to be used for
    the filesystem.

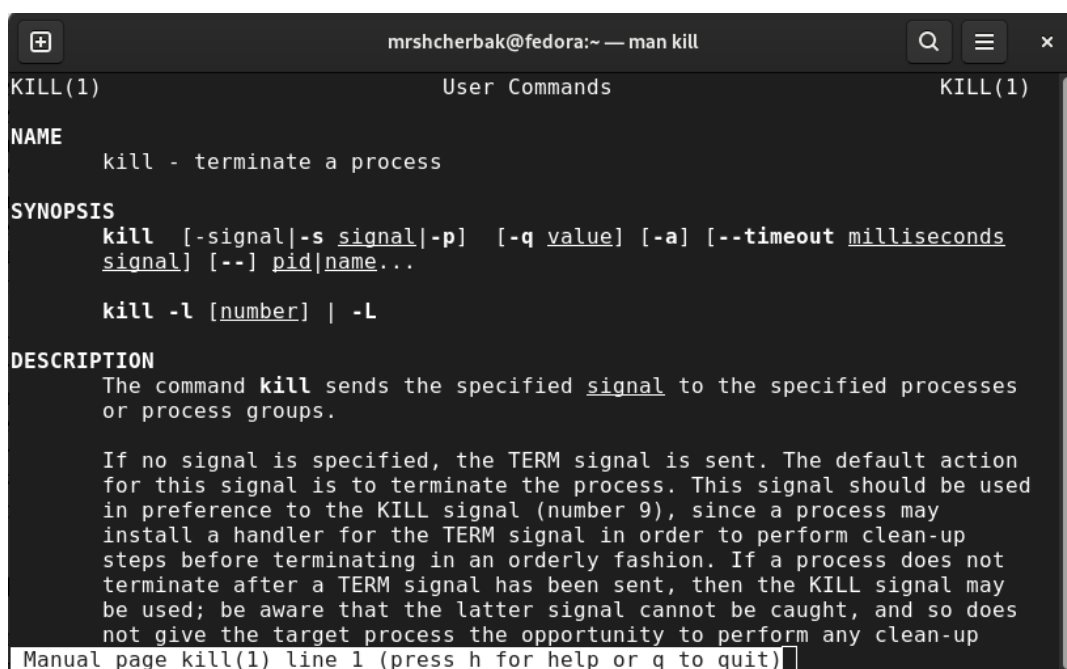
    The exit status returned by mkfs is 0 on success and 1 on failure.

    In actuality, mkfs is simply a front-end for the various filesystem
    builders (mkfs.fstype) available under Linux. The filesystem-specific
    builder is searched for via your PATH environment setting only. Please

Manual page mkfs(8) line 1 (press h for help or q to quit)
```

Рис. 3.13: man mkfs

mkfs позволяет создать файловую систему.



```
KILL(1) User Commands KILL(1)

NAME
    kill - terminate a process

SYNOPSIS
    kill [-signal|-s signal|-p] [-q value] [-a] [--timeout milliseconds
    signal] [--] pid|name...

    kill -l [number] | -L

DESCRIPTION
    The command kill sends the specified signal to the specified processes
    or process groups.

    If no signal is specified, the TERM signal is sent. The default action
    for this signal is to terminate the process. This signal should be used
    in preference to the KILL signal (number 9), since a process may
    install a handler for the TERM signal in order to perform clean-up
    steps before terminating in an orderly fashion. If a process does not
    terminate after a TERM signal has been sent, then the KILL signal may
    be used; be aware that the latter signal cannot be caught, and so does
    not give the target process the opportunity to perform any clean-up

Manual page kill(1) line 1 (press h for help or q to quit)
```

Рис. 3.14: man kill

Для передачи сигналов процессам в Linux используется утилита kill. Ее синтаксис очень прост: kill -сигнал pid\_процесса. Допустим, у нас выполняется утилита ping. Мы хотим ее завершить с помощью kill. Тогда, сначала мы узнаем ее идентификатор с помощью команды ps: ps aux | grep ping. В первой строчке отобразится сама утилита ping, а во второй сама программа ps. Берем нужный PID и завершаем процесс с помощью SIGTERM: kill 20446.

## 4 Контрольные вопросы:

1. Чтобы узнать, какие файловые системы существуют на жёстком диске моего компьютера, использую команду `mount` (Так же можно использовать команды `df -Th`, `cat /etc/fstab`). На моем компьютере есть следующие файловые системы: `devtmpfs`, `tmpfs`, `ext4`, `iso9660`, `btrfs`. `devtmpfs` позволяет ядру создать экземпляр `tmpfs` с именем `devtmpfs` при инициализации ядра, прежде чем регистрируется какое-либо устройство с драйверами. Каждое устройство с майором/минором будет предоставлять узел устройства в `devtmpfs`. `devtmpfs` монтируется на `/dev` и содержит специальные файлы устройств для всех устройств. `tmpfs` – временное файловое хранилище во многих Unix-подобных ОС. Предназначена для монтирования файловой системы, но размещается в ОЗУ вместо ПЗУ. Подобная конструкция является RAM диском. Данная файловая система также предназначена для быстрого и ненадёжного хранения временных данных. Хорошо подходит для `/tmp` и массовой сборки пакетов/образов. Предполагает наличие достаточного объёма виртуальной памяти. Файловая система `tmpfs` предназначена для того, чтобы использовать часть физической памяти сервера как обычный дисковый раздел, в котором можно сохранять данные (чтение и запись). Поскольку данные размещены в памяти, то чтение или запись происходят во много раз быстрее, чем с обычного HDD диска. `ext4` имеет обратную совместимость с предыдущими версиями ФС. Эта версия была выпущена в 2008 году. Является первой ФС из «семейства» Ext, использующая механизм «`extentfile system`», который позволяет добиться меньшей фрагментации

файлов и увеличить общую производительность файловой системы. Кроме того, в Ext4 реализован механизм отложенной записи, который также уменьшает фрагментацию диска и снижает нагрузку на CPU. С другой стороны, хотя механизм отложенной записи и используется во многих ФС, но в силу сложности своей реализации он повышает вероятность утери данных. Характеристики: максимальный размер файла: 16 TB; максимальный размер раздела: 16TB; максимальный размер имени файла: 255 символов. Рекомендации по использованию: наилучший выбор для SSD; наилучшая производительность по сравнению с предыдущими Ext-системами; она так же отлично подходит в качестве файловой системы для серверов баз данных, хотя сама система и моложе Ext3. ISO9660-стандарт, выпущенный Международной организацией по стандартизации, описывающий файловую систему для дисков CD-ROM. Также известен как CDFS (Compact Disc File System). Целью стандарта является обеспечить совместимость носителей под разными операционными системами, такими, как Unix, Mac OS, Windows. Btrfs или B-Tree File System - это совершенно новая файловая система, которая сосредоточена на отказоустойчивости, легкости администрирования и восстановления данных. Файловая система объединяет в себе очень много новых интересных возможностей, таких как размещение на нескольких разделах, поддержка подтомов, изменение размера на лету, создание мгновенных снимков, а также высокая производительность. Но многими пользователями файловая система Btrfs считается нестабильной. Тем не менее, она уже используется как файловая система по умолчанию в OpenSUSE и SUSE Linux.

2. Файловая система Linux/UNIX физически представляет собой пространство раздела диска разбитое на блоки фиксированного размера, кратные размеру сектора – 1024, 2048, 4096 или 8120 байт. Размер блока указывается при создании файловой системы. В файловой структуре Linux имеется один корневой раздел (он же root, корень). Все разделы жесткого диска (если

их несколько) представляют собой структуру подкаталогов, “примонтированных” к определенным каталогам. - “/” – корень. Это главный каталог в системе Linux. По сути, это и есть файловая система Linux. Адреса всех файлов начинаются с корня, а дополнительные разделы, флешки или оптические диски подключаются в папки корневого каталога. Только пользователь root имеет право читать и изменять файлы в этом каталоге. /bin — здесь находятся двоичные исполняемые файлы. Основные общие команды, хранящиеся отдельно от других программ в системе (прим.: pwd, ls, cat, ps); /boot — тут расположены файлы, используемые для загрузки системы (образ initrd, ядро vmlinuz); /dev — в данной директории располагаются файлы устройств (драйверов). С помощью этих файлов можно взаимодействовать с устройствами. К примеру, если это жесткий диск, можно подключить его к файловой системе. В файл принтера же можно написать напрямую и отправить задание на печать; /etc — в этой директории находятся файлы конфигураций программ. Эти файлы позволяют настраивать системы, сервисы, скрипты системных демонов; /home — каталог, аналогичный каталогу Users в Windows. Содержит домашние каталоги учетных записей пользователей (кроме root). При создании нового пользователя здесь создается одноименный каталог с аналогичным именем и хранит личные файлы этого пользователя; /lib — содержит системные библиотеки, с которыми работают программы и модули ядра; /lost+found — содержит файлы, восстановленные после сбоя работы системы. Система проведет проверку после сбоя и найденные файлы можно будет посмотреть в данном каталоге; /media — точка монтирования внешних носителей. Например, когда вы вставляете диск в дисковод, он будет автоматически смонтирован в директорию /media/cdrom; /mnt — точка временного монтирования. Файловые системы подключаемых устройств обычно монтируются в этот каталог для временного использования; /opt — тут расположены дополнительные (необязательные) приложения. Такие программы обычно не подчиняются

принятой иерархии и хранят свои файлы в одном подкаталоге (бинарные, библиотеки, конфигурации); /proc — содержит файлы, хранящие информацию о запущенных процессах и о состоянии ядра ОС; /root — директория, которая содержит файлы и личные настройки суперпользователя; /run — содержит файлы состояния приложений. Например, PID-файлы или UNIX-сокеты; /sbin — аналогично /bin содержит бинарные файлы. Утилиты нужны для настройки и администрирования системы суперпользователем; /srv — содержит файлы сервисов, предоставляемых сервером (прим. FTP или Apache HTTP); /sys — содержит данные непосредственно о системе. Тут можно узнать информацию о ядре, драйверах и устройствах; /tmp — содержит временные файлы. Данные файлы доступны всем пользователям на чтение и запись. Стоит отметить, что данный каталог очищается при перезагрузке; /usr — содержит пользовательские приложения и утилиты второго уровня, используемые пользователями, а не системой. Содержимое доступно только для чтения (кроме root). Каталог имеет вторичную иерархию и похож на корневой; /var — содержит переменные файлы. Имеет подкаталоги, отвечающие за отдельные переменные. Например, логи будут храниться в /var/log, кэш в /var/cache, очереди заданий в /var/spool/ и так далее.

3. Чтобы содержимое некоторой файловой системы было доступно операционной системе необходимо воспользоваться командой mount (монтирование тома).
4. Целостность файловой системы может быть нарушена из-за перебоев в питании, неполадок в оборудовании или из-за некорректного/внезапного выключения компьютера. Чтобы устранить повреждения файловой системы необходимо использовать команду fsck. Также отсутствие синхронизации между образом файловой системы в памяти и ее данными на диске в случае аварийного останова может привести к появлению следующих ошибок: Один блок адресуется несколькими mode (принадлежит несколь-

ким файлам). Блок помечен как свободный, но в то же время занят (на него ссылается `onode`). Блок помечен как занятый, но в то же время свободен (ни один `inode` на него не ссылается). Неправильное число ссылок в `inode` (недостаток или избыток ссылающихся записей в каталогах). Несовпадение между размером файла и суммарным размером адресуемых `inode` блоков. Недопустимые адресуемые блоки (например, расположенные за пределами файловой системы). “Потерянные” файлы (правильные `inode`, на которые не ссылаются записи каталогов). Недопустимые или неразмещенные номера `inode` в записях каталогов.

5. Файловую систему можно создать, используя команду `mkfs`. Ее краткое описание дано в пункте 5) в ходе выполнения заданий лабораторной работы.
6. Для просмотра текстовых файлов существуют следующие команды: `cat`. Задача команды `cat` очень проста –она читает данные из файла или стандартного ввода и выводит их на экран. Синтаксис утилиты: `cat [опции] файл1 файл2 ...`Основные опции: `-b`–нумеровать только непустые строки `-E`–показывать символ `$` в конце каждой строки `-n`–нумеровать все строки `-s`–удалять пустые повторяющиеся строки `-T`–отображать табуляции в виде `^I` `-h`–отобразить справку `-v`–версия утилиты `-nl`. Команда `nl` действует аналогично команде `cat`, но выводит еще и номера строк в столбце слева. `less`. Существенно более развитая команда для пролистывания текста. При чтении данных со стандартного ввода она создает буфер, который позволяет листать текст как вперед, так и назад, а также искать как по направлению к концу, так и по направлению к началу текста. Синтаксис аналогичный синтаксису команды `cat`. Некоторые опции: `-g` –при поиске подсвечивать только текущее найденное слово (по умолчанию подсвечиваются все вхождения) `-N` –показывать номера строк `-head`. Команда `head` выводит начальные строки (по умолчанию – 10) из одного или нескольких документов. Также она может показывать данные, которые передает на вывод другая утилита. Синтаксис аналогичный синтаксису команды `cat`. Основные опции: `-c` (–bytes)



- позволяет задавать количество текста не в строках, а в байтах -n (-lines)
  - показывает заданное количество строк вместо 10, которые выводятся по умолчанию -q (-quiet, -silent) -выводит только текст, недобавляя к нему название файла -v (-verbose) -перед текстом выводит название файла -z (-zero-terminated) -символы перехода на новую строку заменяет символами завершения строк -tail - Эта команда позволяет выводить заданное количество строк с конца файла, а также выводить новые строки в интерактивном режиме. Синтаксис аналогичный синтаксису команды cat. Основные опции:
  - с -выводить указанное количество байт с конца файла -f -обновлять информацию по мере появления новых строк в файле -n -выводить указанное количество строк из конца файла -pid -используется с опцией -f, позволяет завершить работу утилиты, когда завершится указанный процесс -q -не выводить имена файлов -retry- повторять попытки открыть файл, если он недоступен -v-выводить подробную информацию о файле.
7. Утилита `sr` позволяет полностью копировать файлы и директории. Синтаксис: `sr [опции] файл-источник файл-приемник`. После выполнения команды файл источник будет полностью перенесен в файл-приемник. Если в конце указан слэш, файл будет записан в заданную директорию с оригинальным именем. Основные опции: -attributes-only -не копировать содержимое файла, а только флаги доступа и владельца -f, -force -перезаписывать существующие файлы -i, -interactive -спрашивать, нужно ли перезаписывать существующие файлы -L -копировать не символические ссылки, а то, на что они указывают -n -не перезаписывать существующие файлы -P -не следовать символическим ссылкам -r -копировать папку Linux рекурсивно -s -не выполнять копирование файлов в Linux, а создавать символические ссылки -u -скопировать файл, только если он был изменён -x-не выходить за пределы этой файловой системы -p -сохранять владельца, временные метки и флаги доступа при копировании -t -считать файл-приемник директорией и копировать файл-источник в эту директорию.

8. Команда `mv` используется для перемещения одного или нескольких файлов (или директорий) в другую директорию, а также для переименования файлов и директорий. Синтаксис: `mv [-опции] старый_файл новый_файл`. Основные опции: `-help` – выводит на экран официальную документацию об утилите `-version` – отображает версию `mv` `-b` – создает копию файлов, которые были перемещены или перезаписаны `-f` – при активации не будет спрашивать разрешение у владельца файла, если речь идет о перемещении или переименовании файла

`-i` – наоборот, будет спрашивать разрешение у владельца `-n` – отключает перезапись уже существующих объектов `-strip-trailing-slashes` – удаляет завершающий символ / у файла при его наличии `-t [директория]` – перемещает все файлы в указанную директорию `-u` – осуществляет перемещение только в том случае, если исходный файл новее объекта назначения

`-v` – отображает сведения о каждом элементе во время обработки команды. Команда `rename` также предназначена, чтобы переименовать файл. Синтаксис: `rename [опции] старое_имя новое_имя файлы`.

Основные опции: `-v` – вывести список обработанных файлов `-n` – тестовый режим, на самом деле никакие действия выполнены не будут `-f` – принудительно перезаписывать существующие файлы.

9. Права доступа – совокупность правил, регламентирующих порядок и условия доступа субъекта к объектам информационной системы (информации, её носителям, процессам и другим ресурсам) установленных правовыми документами или собственником, владельцем информации. Права доступа к файлу или каталогу можно изменить, воспользовавшись командой `chmod`. Сделать это может владелец файла (или каталога) или пользователь с правами администратора.

Синтаксис команды: `chmod режим имя_файла`. Режим имеет следующие компоненты структуры и способ записи: `=` установить право. `-` лишить права. `+` дать право. `r` чтение. `w` запись. `x` выполнение. `u` (user) владелец файла.

g (group) группа, к которой принадлежит владелец файла. o (others) все остальные.

## **5 Выводы:**

Таким образом, в ходе ЛРН<sup>№5</sup> я ознакомилась с файловой системой Linux, её структурой, именами и содержанием каталогов. Приобрела практические навыки по применению команд для работы с файлами и каталогами, по управлению процессами (и работами), по проверке использования диска и обслуживанию файловой системы.